Project Llama

Making A Cappella Arranging Easier

Jason Lu

Spring 2015

Table of Contents

Introduction

A cappella is the art of making music with just the human voice. It has gained popularity over the past few years, mostly in part of the movie *Pitch Perfect*. A cappella groups are fun to listen to because they take popular songs on the radio but perform them only with the voice. From young high school co-ed groups like Vocal Rush to established professional male groups like Straight No Chaser, you are bound to find groups of all sorts of ages, genders, and genres. The community for a cappella is huge and the only way to stand out from the rest is through unique arrangements.

To give an idea of how big the a cappella community is, there were over three hundred collegiate groups competing in this year's ICCAs (International Competition of Collegiate A cappella)[1]. According to Deke Sharon, director of the Contemporary A Cappella Society of America, the number of college groups has quadrupled over the last ten years[2].

While the community is ever growing, the tools available for arranging a cappella music are still limited. Arranging is defined as reconceptualizing of a previously composed work. It basically means to listen to a piece and transcribing it into sheet music. In terms of a cappella, this means taking a song on the radio and writing vocal parts. To make an a cappella arrangement these days, you have to use music notation software such as Noteflight or Sibelius. The process is slow and tedious which usually translates to fewer arrangement output per year. It also relies heavily on the arranger knowing some music theory including reading/writing music, chord structure, and voice leading. It is challenging to arrange because you have to account for the different voice types and the different talents in your group. No two a cappella groups are the same which makes it that much more challenging. In addition, arrangements need to be unique in order for your group to stand out.

---

[1] Varsity Vocals: http://varsityvocals.com/2015-icca-lineup-announced/

[2] Sandman, Victor. "Boy bands over bach." *American Music Teacher* Apr.-May 2005: 40+. *Academic OneFile*. Web. 16 June 2015.

Arranging expertise is an essential skill for a cappella groups, according to music teacher Heidi Welch[3].

With the need of music theory knowledge for current arrangements, that means only a select few per group are capable of producing a cappella music. It is not uncommon for many singers to have little music theory background. Choirs spend very little time teaching music theory anyway. Music reading, an important music theory concept, is not even taught for most choirs. In a study done in 1961, it was found that only 37% of the 244 high school groups taught music reading[4]. But that does not mean only a select few have great ideas. What if there was a way to give everyone else a chance to show their arranging ideas?

I have been arranging for three years for my group That's The Key, making over ten arrangements ranging from genres like Rap to Country. In those ten arrangements, I have spent as little as 45 minutes to as much as five hours working on a single arrangement. Also, in my group of 17 people, only four arranged for the group but many had great ideas. Arranging a cappella is too limiting in its current state. My project aimed to fix this in many ways. I wanted to break the barrier between arranging and those who have great ideas but do not have the necessary skills.

[3] Poliniak, Susan. "Contemporary a cappella as a show choir alternative." *Teaching*. *OneFile*. Web. 16 June 2015.

4Sightsinging in the Secondary Choral Ensemble: A Review of the Research

Steven M. Demorest

Bulletin of the Council for Research in Music Education

No. 137 (Summer, 1998) , pp. 1-15

Published by: University of Illinois Press on behalf of the Council for Research in Music Education

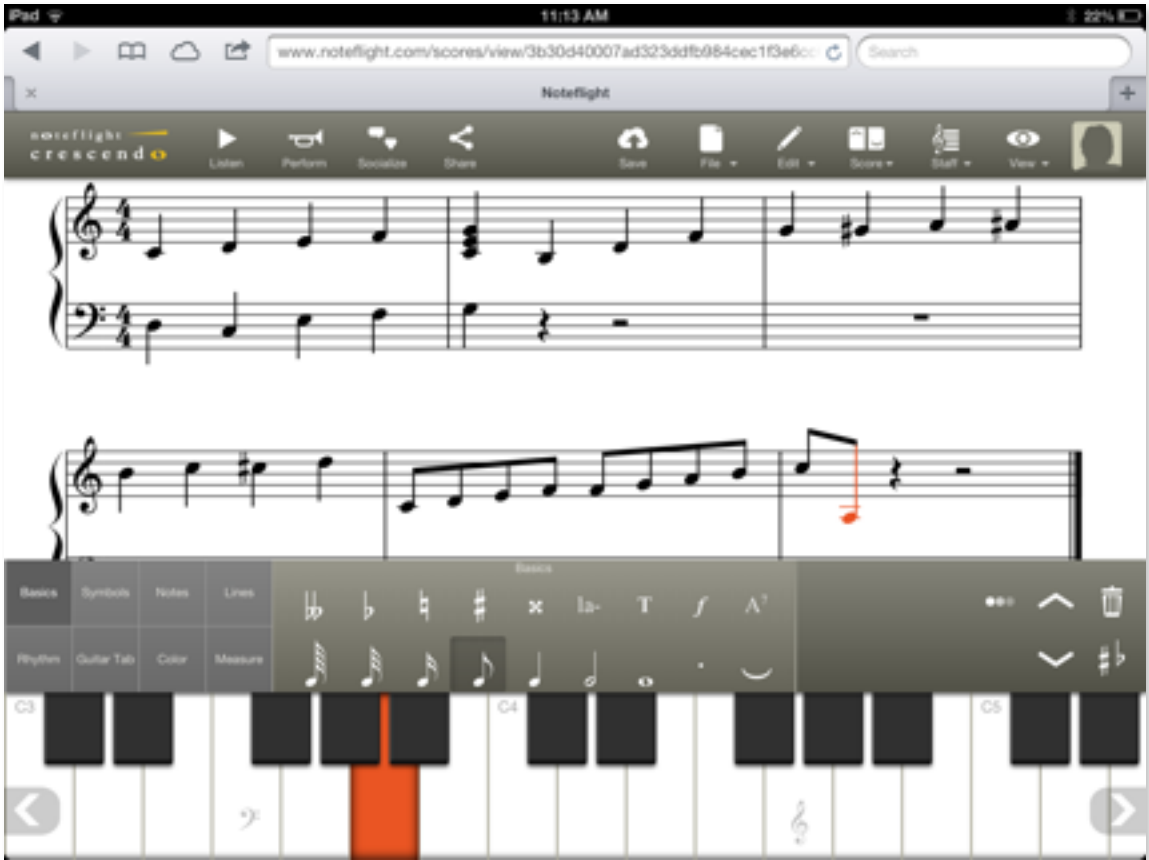
Stable URL: http://www.jstor.org/stable/40318928

Deliverable

Project Llama aimed to be a tool that the a cappella community can find helpful. It would streamline the process of making arrangements by recording parts using only the voice instead of relying on the mouse and keyboard. Llama was an iOS app that allowed users to record their arrangements vocally rather than tediously writing notes in notation software. This hopefully increased the amount of a cappella members who can arrange as well as increase the amount of arrangements per group.

Literature and Technology Review

Noteflight, Sibelius, and Finale are all types of music notation software. Users can click and drag notes around to write sheet music. They also come with a MIDI player so you can hear what you have written instantaneously. Modern composers and arrangers use these programs heavily as they are the industry standard with Sibelius being the most popular. These programs are extremely powerful in the sense that they offer many features for composers and arrangers. Unfortunately, you are required to understand many music theory concepts if you



want to use these programs to their full extent (i.e. chord structures, voice leading, rhythms).

I used Noteflight for all of my arrangements and I think it is a really convenient program, but it also took a very long time to work on my songs. The end product is very good, though, because everyone can learn by reading the sheet music or by listening to the MIDI player.

Another option is the use of Digital Audio Workstations such as ProTools, GarageBand, and Logic Pro. These programs are primarily used in the industry to record and produce songs. Using these programs would shorten arranging time too, but these programs can cost as much as $500. If you are using these programs, you might as well be an audio engineer as well.

Finally, there is Loopy. Loopy is a loop station application for iOS that was recently made popular by Jimmy Fallon. You essentially record "loops" that repeat

forever and other "loops" over them. This is a great idea for arranging because you can build harmonies easily with just your voice. However, you are limited to just one set of loops for one chord progression. Bridges of songs are usually different chord progressions which can not be easily done in Loopy.

I decided to model my app off of Loopy but with the a cappella model in mind. I wanted the simplicity of recording parts that Loopy had but with the flexibility that Noteflight offered. My project would only focus on the human voice, much like Loopy does but without repeating. I wanted to have recordable blocks that could be moved and have a fully recorded arrangement as a result. There are no notes to write, only the audio that the user will hear at the end. By doing this, it eliminates the need for music theory or shelling out hundreds of dollars and puts the tools right on your phone.

Technology Overview

This app was written in Apple's new language: Swift. I chose Swift because it is the new standard iOS/OS X language as of a year ago. This is a way for me to future-proof the project as I will not have to learn another language for a long time. I also included the AVFoundation API (Apple's standard API for audio and video) and SpriteKit for the visual effects.

From their description, AVFoundation is "one of several frameworks that you can use to play and create time-based audiovisual media." It is one of the standard frameworks in the OS X and iOS family. I chose to use AVFoundation because there is a lot of documentation surrounding the framework and there are many tutorials for it online. My only gripe was that most of it is has not been documented for Swift yet.

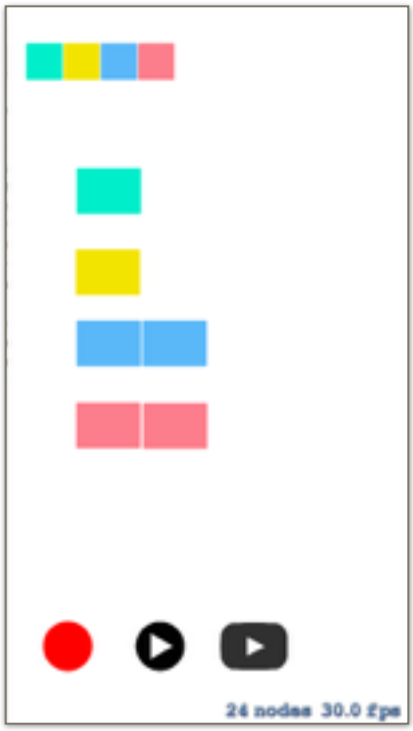
SpriteKit is primarily used for making games but I found it to be appropriate to represent my block idea. The API allows for changing of the position of sprites

efficiently. There are also useful physics functions which makes it perfect for game developers. For my project, SpriteKit allowed me to create the block nodes for me to drag around the screen easily.

I chose iOS for development because there are more people with iPhones in my group. I also own an iPhone which made testing easy and convenient. I do plan on making an Android version later.

Design and Implementation

I went through several designs for this project. My initial idea was to have multiple blocks that can be rearranged and the audio would follow the order. This idea turned out to be very complicated to implement so I decided to shelf it for later versions.



Instead, I simplified it to just four recordable blocks. This was much easier to implement as I only had make four audio recorders and audio players. I also removed the record and play buttons and just made the blocks buttons. I think this was an easier-to-understand design. This gave it more of a "Loopy" feel as you are recording bits, except they do not repeat. For my research, this version was the one that the test

subjects used:

SpriteKit is split up into scenes and since I only have one scene, most of my code was inside one class. I set up responsive labels (SKLabelNode) and blocks as SKShapeNodes. The Play All button is a SKSpriteNode using an image that I found on the internet.

For recording and playback, I used instances of theAVAudioRecorder and AVAudioPlayer. I made four separate instances, one for each part/block. The AVAudioRecorder takes in a URL and need to be primed before they can record. For the audio settings, I did a standard 16 bit rate, 2 channels, and 44.1 kHz.

```
let recordSettings =
[AVEncoderAudioQualityKey: AVAudioQuality.Min.rawValue,
    AVEncoderBitRateKey: 16,
    AVNumberOfChannelsKey: 2,
    AVSampleRateKey: 44100.0]

let dirPaths =
NSSearchPathForDirectoriesInDomains(.DocumentDirectory,
    .UserDomainMask, true)
let docsDir = dirPaths[0] as! String
let sopFilePath = docsDir.stringByAppendingPathComponent("sop.caf")
let altFilePath = docsDir.stringByAppendingPathComponent("alt.caf")
let tenFilePath = docsDir.stringByAppendingPathComponent("ten.caf")
let basFilePath = docsDir.stringByAppendingPathComponent("bas.caf")
let sopFileURL = NSURL(fileURLWithPath: sopFilePath)
let altFileURL = NSURL(fileURLWithPath: altFilePath)
let tenFileURL = NSURL(fileURLWithPath: tenFilePath)
let basFileURL = NSURL(fileURLWithPath: basFilePath)
var error: NSError?

if let err = error {
    println("audioSession error: \(err.localizedDescription)")
}


sopRecord = AVAudioRecorder(URL: sopFileURL, settings: recordSettings as
    [NSObject : AnyObject], error: &error)
altRecord = AVAudioRecorder(URL: altFileURL, settings: recordSettings as
    [NSObject : AnyObject], error: &error)
tenRecord = AVAudioRecorder(URL: tenFileURL, settings: recordSettings as
    [NSObject : AnyObject], error: &error)
basRecord = AVAudioRecorder(URL: basFileURL, settings: recordSettings as
    [NSObject : AnyObject], error: &error)

if let err = error {
    println("audioSession error: \(err.localizedDescription)")
} else {
    sopRecord?.prepareToRecord()
    altRecord?.prepareToRecord()
    tenRecord?.prepareToRecord()
    basRecord?.prepareToRecord()
}
```

Each block also had a label that would display whether it was recording or not. The touch events were set up in the touchesBegan class where I made functions that were called depending on the location of the touch. If the touch location was inside a certain block's location, then it would start the recording and change the label accordingly.

```
for touch: AnyObject in touches {

    let location = touch.locationInNode(self)

    if sopButton!.containsPoint(location) {
        if sopRecord?.recording == false {
            sopLabel.text = "Recording!"
            sopRecord?.record()
        }
        else {
            sopRecord?.stop()
            sopLabel.text = "Done"
        }
    }
```

Next, the audio played back simultaneously through the use of the AVAudioPlayers. These take in the url of the audio set by the AVAudioRecorders

and play when you call the play function. I set all four to play at the same time when the Play All button is pressed.

```
if playAll!.containsPoint(location) {
    var error: NSError?
    sopPlayer = AVAudioPlayer(contentsOfURL: sopRecord?.url, error: &error)
    tenPlayer = AVAudioPlayer(contentsOfURL: tenRecord?.url, error: &error)
    altPlayer = AVAudioPlayer(contentsOfURL: altRecord?.url, error: &error)
    basPlayer = AVAudioPlayer(contentsOfURL: basRecord?.url, error: &error)

    sopPlayer?.delegate = self
    altPlayer?.delegate = self
    tenPlayer?.delegate = self
    basPlayer?.delegate = self


    audioSession.setCategory(AVAudioSessionCategoryPlayAndRecord,
        error: &error)
    audioSession.overrideOutputAudioPort(AVAudioSessionPortOverride.Speaker,
        error: nil)


    if let err = error {
        println("audioPlayer error: \(err.localizedDescription)")
    }
    else {
        sopPlayer?.play()
        altPlayer?.play()
        tenPlayer?.play()
        basPlayer?.play()
    }
}
```

Finally, the metronome is simply a circle shape that hides and unhides based on a time delay. I use a sequence of actions (SKAction) separated by a time delay (waitForDuration). The starting time delay is half a second and are changed by 0.7 seconds depending on which arrow is pressed. I decided to make the metronome purely visual as I did not want to hear the clicks in the final recording.

```
func startBlink(stdTime: Double) {
    let blink = SKAction.sequence([
        SKAction.waitForDuration(stdTime),
        SKAction.hide(),
        SKAction.waitForDuration(stdTime),
        SKAction.unhide()])

    let blinkForever = SKAction.repeatActionForever(blink)

    metronome!.removeActionForKey("blink")

    metronome!.runAction(blinkForever, withKey: "blink")
}
```

Analysis and Verification of Success

For this project, I wanted to know a few things: is my project a significant contribution to the a cappella community, is my project significant to the music community as a whole, and if my project was easy to use. My goal was to interview many a cappella members and musicians on campus. I wanted to obtain certain average scores based on a ranking that I supplied in survey. My target marks were as follows:

Grade	C	B	A
Significance to the A Cappella Community	3.5	4	4.5
Significance to the Music Community	3	3.5	4
Usability	4	4.5	5

I stressed more importance for usability because I wanted my project to be easier to use than Noteflight or other similar programs.

For my IRB/Human Subjects Committee approved study, I asked singers to think of a song and to make an a cappella version of that song using my app. After they were done, they were handed a survey with these questions:

Do you know how to read and write music?

Have you used other music notation software (i.e. Noteflight, Sibelius, Finale)? If so, please list the one(s) you have used.

Do you play an instrument other than voice? If so, please list them.

On the following scale: How easy was it to create a song using this app? (1 - very difficult, 5 - very easy)

On the following scale: How would you rank this application's relevance to the a cappella community? (1 - not very relevant, 5 - very relevant)

Do you think this app, in its current state, can serve as an important tool for the a cappella community?

Do you think this app, in its current state, can serve as an important tool for other musicians?

It was important for me to ask if they knew how to read and write music because I wanted to see if people without a music theory background had a positive experience.

Results

Significance to A Cappella Community	**4.6**	**A**
Significance to Music Community	**3.1**	**C+**
Usability	**4.4**	**B-**

I ended up recruiting about twelve a cappella members total from both Take It SLO and That's The Key, and recruited four music majors who were hanging out at in the music lobby. After interviewing, I got some decent results. I received a 4.6 for significance to the a cappella community. This is great because it means that my idea is a good one and is significant. I am on the right track. I also got a 3.1 for the music community significance, and I am not upset about that. I did not intend for my project to help other musicians so that category scored reasonably low. Finally, I scored a 4.4 for usability. This makes sense because I did not have many features in the test version of the app. One of the biggest complaints I got in my surveys was the need for a metronome. Most of the singers had trouble keeping time with their recordings. I later added this functionality.

I got some great feedback from the surveys. An important trend was the positive responses from those who have no music theory background. Of the twelve a cappella members I surveyed, about a third of them did not have a background in music theory. One such singer said, "It seems like it would be super useful for those who don't quite know how to read and write music but have good ideas." This validates my claim that my project can be helpful for those without a strong music background.

Overall, I got very positive responses from the a cappella members. The main criticisms were lack of features, which I addressed after the survey. I do plan to bring more of the features that they wanted.

Societal Impacts

I hope that my project can be a significant contribution to the a cappella community. It was my goal to create a new tool that a cappella singers can use, and it seems like I am on the right track. If groups around the world can benefit from this project then I am satisfied. Society can always use more a cappella songs! I hope this project also makes arranging fun instead of the tedious process it is right now.

Future Work

I plan to continue adding features to this project, especially my initial design idea. I definitely want to include the moveable block design to allow for more audio recordings. I also plan to make this project open source so others can contribute because I am sure that there are others out there who have great ideas for this. There will hopefully be an Android version in the near future as well.

References

AVFoundation: https://developer.apple.com/av-foundation/

Loopy: http://loopyapp.com

Noteflight: https://www.noteflight.com

Poliniak, Susan. "Contemporary a cappella as a show choir alternative." *Teaching*

Music Jan. 2013: 52+. *Academic OneFile*. Web. 16 June 2015.

Sandman, Victor. "Boy bands over bach." *American Music Teacher* Apr.-May 2005:

40+. *Academic OneFile*. Web. 16 June 2015.

Sightsinging in the Secondary Choral Ensemble: A Review of the Research
Steven M. Demorest
Bulletin of the Council for Research in Music Education
No. 137 (Summer, 1998) , pp. 1-15
Published by: University of Illinois Press on behalf of the Council for Research in Music Education
Stable URL: http://www.jstor.org/stable/40318928

SpriteKit: https://developer.apple.com/library/ios/documentation/

GraphicsAnimation/Conceptual/SpriteKit_PG/Introduction/Introduction.html

Swift: https://developer.apple.com/swift/